

MODÜL 11

İLERİ SEVİYE VERİ YAPILARI



Şekil 11.1: Bölümle ilgili örnek uygulamalara karekoddan ulaşabilirsiniz.

11.1. Sayıların İleri Seviye Özellikleri

Bu bölümde sayı veri tipleri derinlemesine incelenecek, ayrıca sayılarla ilgili bazı fonksiyonlar anlatılacaktır.

Onluk tabandaki bir sayıyı ikilik tabana çevirme:

10'luk sistemdeki sayıları 2'lik sistemdeki bir sayıya çevirmek için **bin()** fonksiyonu kullanılır.

Aşağıdaki örnek kodları etkileşimli kabuk üzerinde çalıştırınız.

MODÜL 11

```
bin(5)
>>> '0b101'
```

Buradaki **0b** ifadesi sayının ikilik tabanda olduğunu belirtmektedir. 101 ifadesi ise sağdan itibaren “iki üzeri sıfır çarpı 1 + iki üzeri bir çarpı 1 + iki üzeri iki çarpı 1” şeklinde tanımlanmaktadır.” Bu işlemin sonucu $(2^0*1) + (2^1*0) + (2^2*1) = 5$ olarak hesaplanmaktadır.

```
bin(12)
>>> '0b1100'
```

$(2^0*0) + (2^1*0) + (2^2*1) + (2^3*1) = 12$
şeklinde hesaplanmıştır.

Onluk tabandaki bir sayıyı onaltılık tabana çevirme:

10'luk sayı sistemdeki sayıları 16'lık sistemdeki bir sayıya çevirmek için **hex()** fonksiyonu kullanılır.

```
hex(18)
>>> '0x12'
```

Buradaki **0x** ifadesi sayının 16'lık sistemde olduğunu göstermektedir.

```
hex(18)
>>> '0x12'
```

$(16^0*2) + (16^1*2) = 2 + 16 = 18$
olarak hesaplanır.

11.2. Sayılar Üzerinde Uygulanabilen Fonksiyonlar

abs() Fonksiyonu: İngilizcede ‘mutlak’ anlamına gelen **absolute** adlı bir sözcüğünün kısaltmasıdır. **abs()** fonksiyonu bir sayının mutlak değerini almak için kullanılır. Fonksiyon tek parametre olarak sayının mutlak değerini alır.

```
abs(-10)
```

```
>>> 10
```

```
abs(15)
```

```
>>> 15
```

```
abs(-3.2)
```

```
>>> 3.2
```

round() Fonksiyonu: **round()** fonksiyonu bir sayıyı belirli kriterlere göre yukarı veya aşağı yuvarlamak için kullanılır.

```
round(5.7)
```

```
>>> 6
```

```
round(5.3)
```

```
>>> 5
```

```
round(5.5)
```

```
>>> 6
```

Eğer girilen değer virgülden sonraki kısmı 5 ve üzerinde bir değerse yukarı, altında bir değerse aşağı yuvarlama işlemi yapar.

MODÜL 11

chr() Fonksiyonu: chr() fonksiyonu, kendisine parametre olarak verilen bir tam sayının karakter olarak (ASCII) karşılığını verir.

```
chr(65)
>>> 'A'

chr(100)
>>> 'd'
```

max() Fonksiyonu: max() fonksiyonu, bir dizi içindeki sayıların en büyüğünü verir.

```
max(3,5,7,8,9)
>>> 9

liste=[3,8,2,6,15]
max(liste)
>>> 15
```

min() Fonksiyonu: min() fonksiyonu, max() fonksiyonunun tam tersi işlem yapar.

```
min(4,7,1,2,9)
>>> 1

liste=[4,7,6,9,3]
min(liste)
>>> 3
```

pow() Fonksiyonu: pow() fonksiyonu power sözcüğünün kısaltmasında türetilmiştir. Bir sayının üssünü almak için kullanılır. İki adet parametre alır, birinci parametre üssü alınacak sayıyı, ikinci parametre ise kuvvetini ifade eder.

```
pow(3,4)
>>> 81
pow(9,0.5)
>>> 3.0
```

sum() Fonksiyonu: `sum()` fonksiyonu, dizi içerisindeki değerlerin toplamını bulmamızı sağlar. `sum()` fonksiyonuna girilen değerler liste ya da tuple türünden olması gerekmektedir.

```
sum([5,3,8,6])
>>> 22
```

11.3. İleri Seviye Karakter Dizileri (String)

Karakter dizilerinin özel metotları:

replace() Fonksiyonu: `replace()` fonksiyonu, bir karakter dizisi içindeki karakterleri başka karakterlerle değiştirmeyi sağlar. **replace()** fonksiyonu iki adet parametre alır. Birinci parametre değişecek karakter ya da karakterleri, ikinci parametre ise yerine gelecek karakter ya da karakterleri ifade eder.

```
a="python"
a.replace("p","P")
>>> 'Python'
```

split() Fonksiyonu: `split()` fonksiyonu bir karakter dizisini verilen kurala göre bölme işlemi yapar. Eğer `split()` fonksiyonuna parametre verilmezse boşluk karakterine göre yapar.

```
a="Milli Eğitim Bakanlığı"
a.split()
>>> ['Milli', 'Eğitim', 'Bakanlığı']
```

MODÜL 11

Ya da belirli kriterlere göre bölme işlemi de yapılabilir.

```
b="T.B.M.M"  
b.split(".")  
>>> ['T', 'B', 'M', 'M']
```

upper() ve lower() fonksiyonları: Bu fonksiyonlar karakter dizilerini büyük veya küçük harfe çevirme işlemi yapar.

```
"Merhaba dünya".upper()  
>>> 'MERHABA DÜNYA'  
  
"Merhaba DÜNYA".lower()  
>>> 'merhaba dünya'
```

join() fonksiyonu: **split()** fonksiyonunun tam tersi işlem yapar. Liste içerisinde bulunan karakter dizilerini verilen kurala göre birleştirmek için kullanılır.

```
"-".join(["Merhaba", "Dünya"])  
>>> 'Merhaba-Dünya'
```

capitalize fonksiyonu: **capitalize()** fonksiyonu karakter dizilerinin sadece ilk harfini büyük yapmak için kullanılır.

```
a="python programlama dili"  
a.capitalize()  
>>> 'Python programlama dili'
```

find() fonksiyonu: find() fonksiyonu karakter dizisi içerisindeki bir karakterin konumunu sorgular. Bulduğu ilk değeri döndürür.

```
a="armağan"  
a.find("a")  
>>> 0
```

İlk bulunduğu indis değerini verir.

rfind() fonksiyonu: find() fonksiyonun benzeri işlemi yapar ancak arama işlemini sağ taraftan başlayarak yapar.

```
a="armağan"  
a.rfind("a")  
>>> 5
```

isdigit() fonksiyonu: isdigit() fonksiyonu, karakter dizisinin bir sayısal değer olup olmadığını kontrol eder. Eğer karakter dizisindeki tüm karakterler rakamdan oluşuyorsa True, değilse False değeri döndürür.

```
metin = "12345"  
print(metin.isdigit())  
>>> True  
  
metin = "asd123"  
print(metin.isdigit())  
>>> False
```




Geniş Kütüphane
Web Uygulamaları
Açık Kaynak
Makine Öğrenimi Uygulamaları
Modüler Yapı
Yapay Zeka Uygulamaları
Çapraz Platform Uyumlu
Güçlü ve Dinamik

